SCHOOLHITS

PROJECT REPORT

ON

SCHOOLHITS

BY

LABADE VIJAY
BHOYATE AKSHAY
PATIL GANESH
SURYAWANSHI TANUJA
GAIKWAD SONALI

SAVITRIBAI PHULE PUNE UNIVERSITY
MASTER OF COMPUTER APPLICATION

ASM's INSTITUTE
PUNE-411019

2021-22

INDEX

1. INTRODUCTION:

- The title of the project is “SCHOOLHITS”.
- This project will handle whole the information of Teacher, Student and Employees.
- We will create the information form of the student, teacher and employees.
- In this system the data can handle easily by the admin.
- The admin can save the data, delete the data, update the data and change the data.

1.1 Existing System:

The existing system is majorly clerical process this is not computerized system.

There are some drawbacks in existing system:

- The existing system performs the task as manually.
- In this system there are lots of paper works.
- The existing system is very time consuming processes.
- It is Difficult to generate reports.
- There is difficulty in storing and retrieving the data.
- It has a large amount of repeated data.
- There is lots of man power required. Due to the above drawback, the existing system is very complicated and not secure system.

1.2 Scope:

- This system is aimed at total user-friendly as well as efficient management of varied task.
- These task may range from registering the data of new student, teacher and employees.
- This system have the essential features necessary for making the administrative division of school effective.
- The software will display view the information of the teacher, student and employees.

1.3 Operating Environment – Hardware and Software :

- **CLIENT SIDE:**

Operating System	Windows 10
Processor	Intel-Pentium Dual Core
RAM	1 GB RAM
Browser	Internet Explorer, Google Chrome, Mozilla firefox, etc.

- **SERVER SIDE:**

Operating system	Windows 10
Front End	Java, Swing
Connectivity	JDBC Drivers, Php My admin
Backend	MySQL Database Server: Xampp

1.4 Technology Used:

- Frond end: HTML, CSS, JavaScript.
- Back end: Java
- Database: MySQL
- Framework: NetBeans 12.6
- Server: XAMPP

2.2.1 PROPOSED SYSTEM:

- The aim of proposed system is to develop a system of improved facilities.
- The proposed system can overcome all the limitations of the existing system.
- The system provides proper security and reduces the manual work.
- Security of data
- Ensure data accuracy
- Proper control of the higher officials.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive
- Minimum time required
- Fast access to database
- Less error
- More storage capacity.
- Search facility

2.1 Objectives of System:

- The main objective of the school management system is to manage the data of the school and the details of the teacher, student and employees.

- The school management system is user friendly system which manages all the information about school teachers, students and employees.

- The project is totally built at administrative end and thus only the administrator is guaranteed the access.

- Only the admin have the access of the system, admin can change the data, save the data, update the data and delete the data. And also search the data of specific teacher, student and employees.

3.FEASIBILITY STUDY:

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system.

3.1 Technical Feasibility:

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system.

3.2 Economic Feasibility:

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility

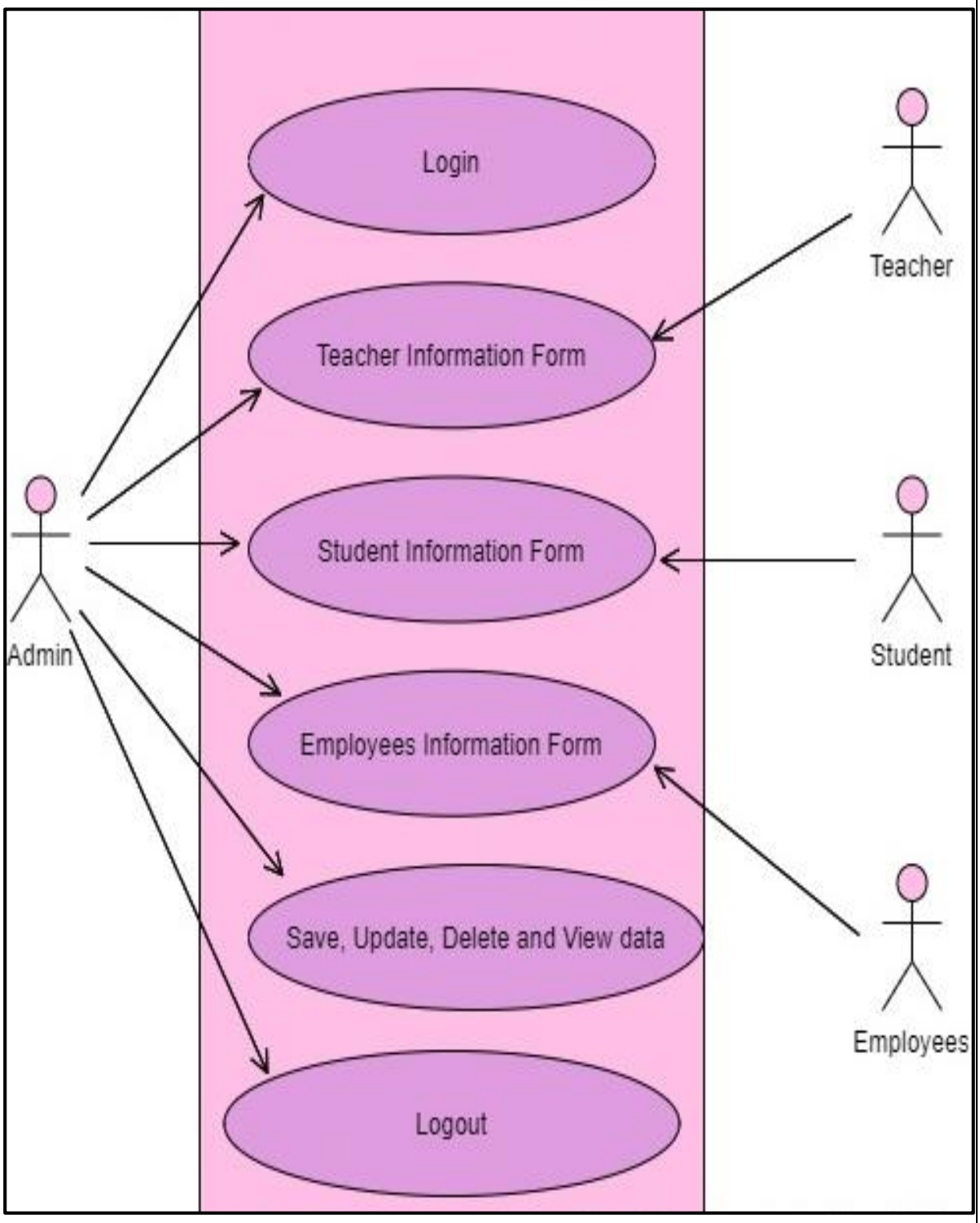
assessment is to determine the positive economic benefits to the organization that the proposed system will provide.

3.3 Operational Feasibility:

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters.

4. ANALYSIS & DESIGN:

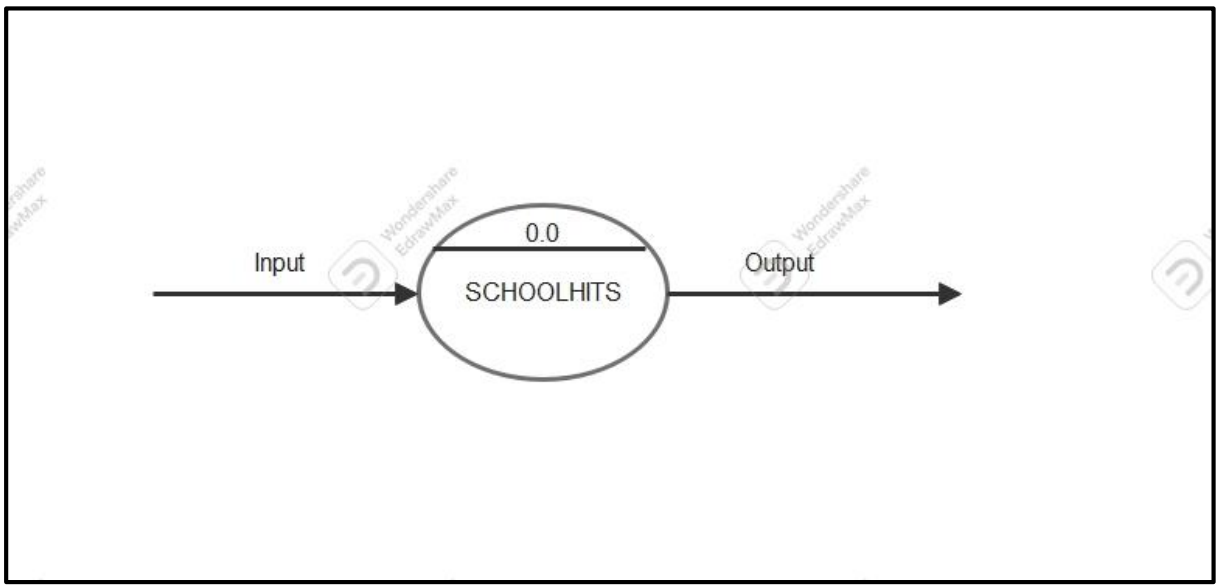
4.1 Use Case Diagram:



4.1 Use Case Diagram

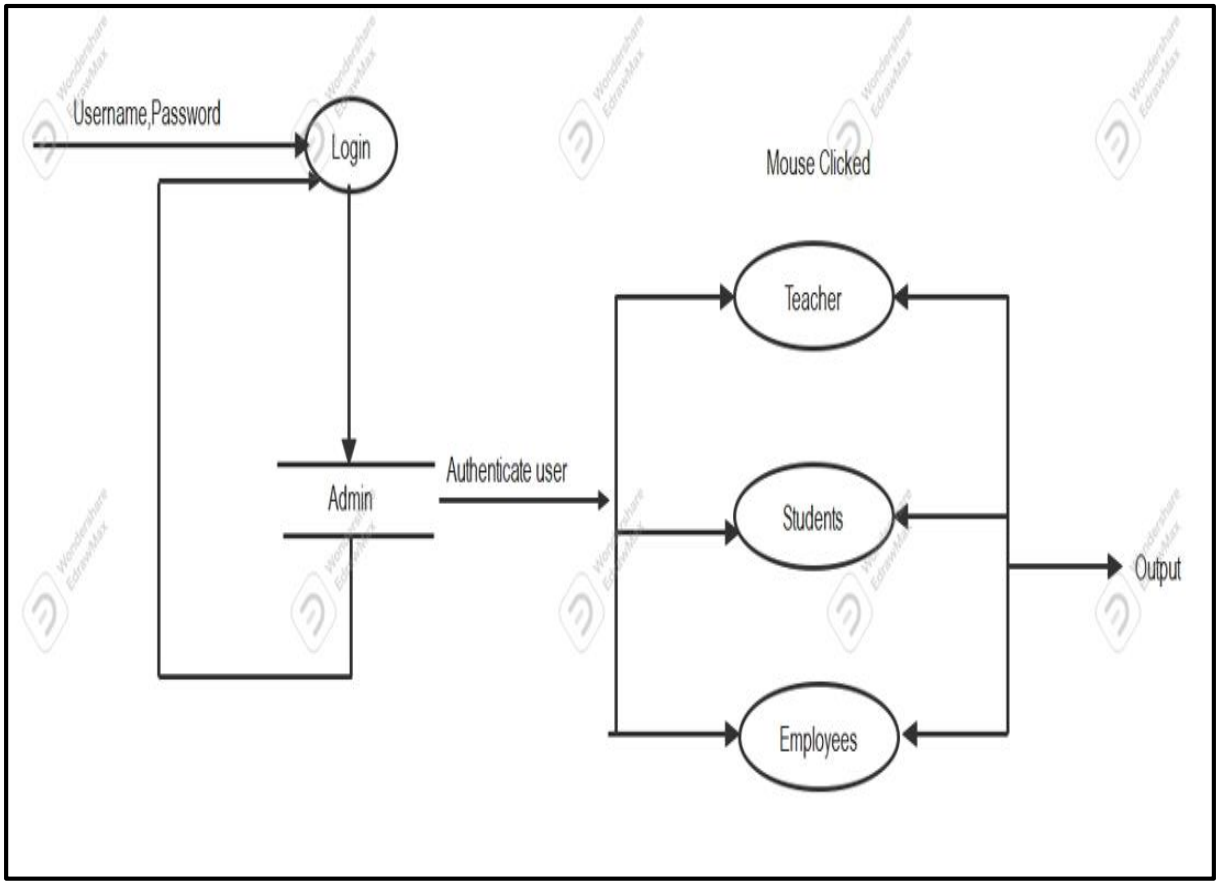
4.2 DATA FLOW DIAGRAM: ZERO LEVEL DFD :

1. ZERO LEVEL DFD:



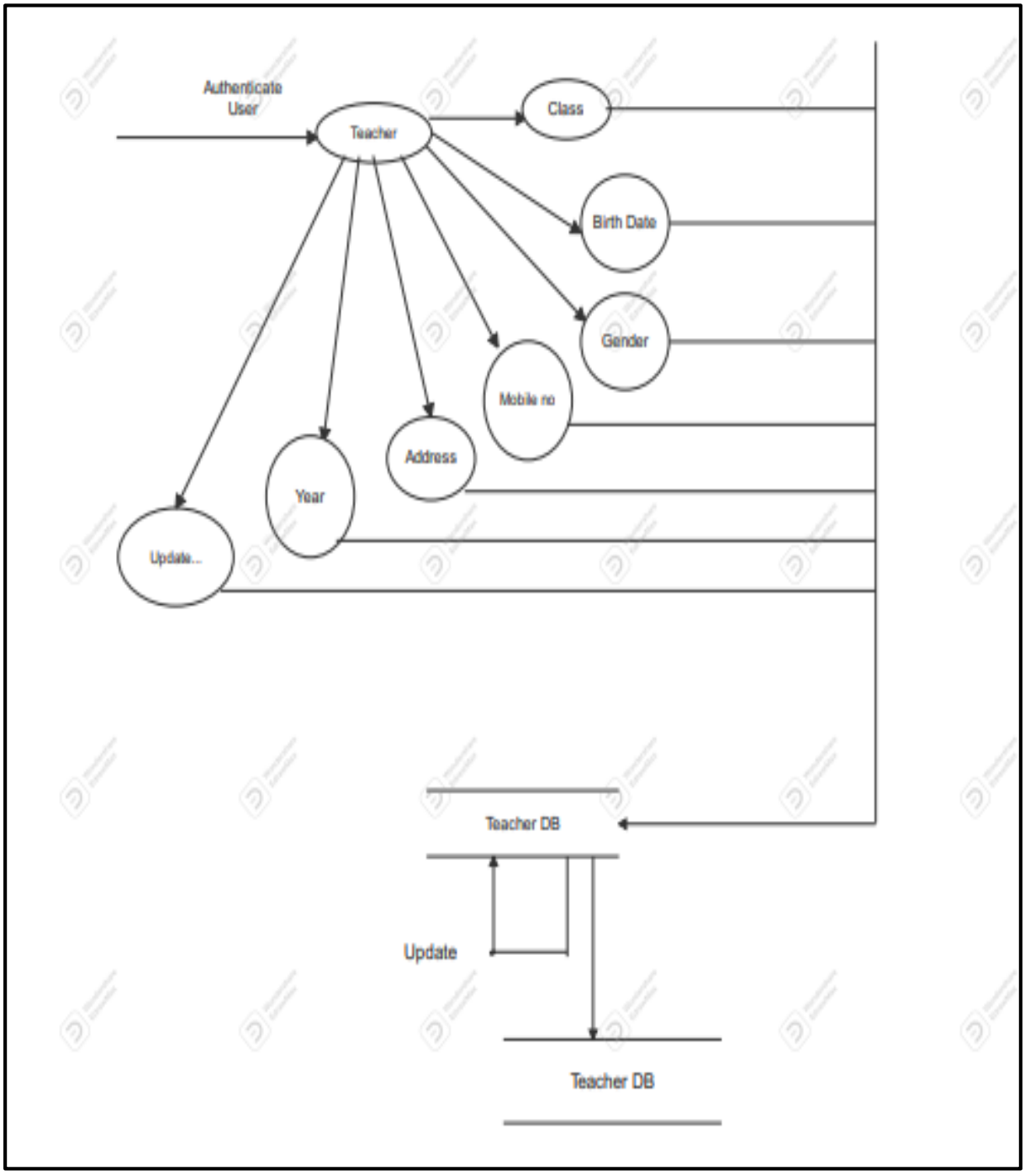
ZERO LEVEL DFD

1. FIRST LEVEL DFD:



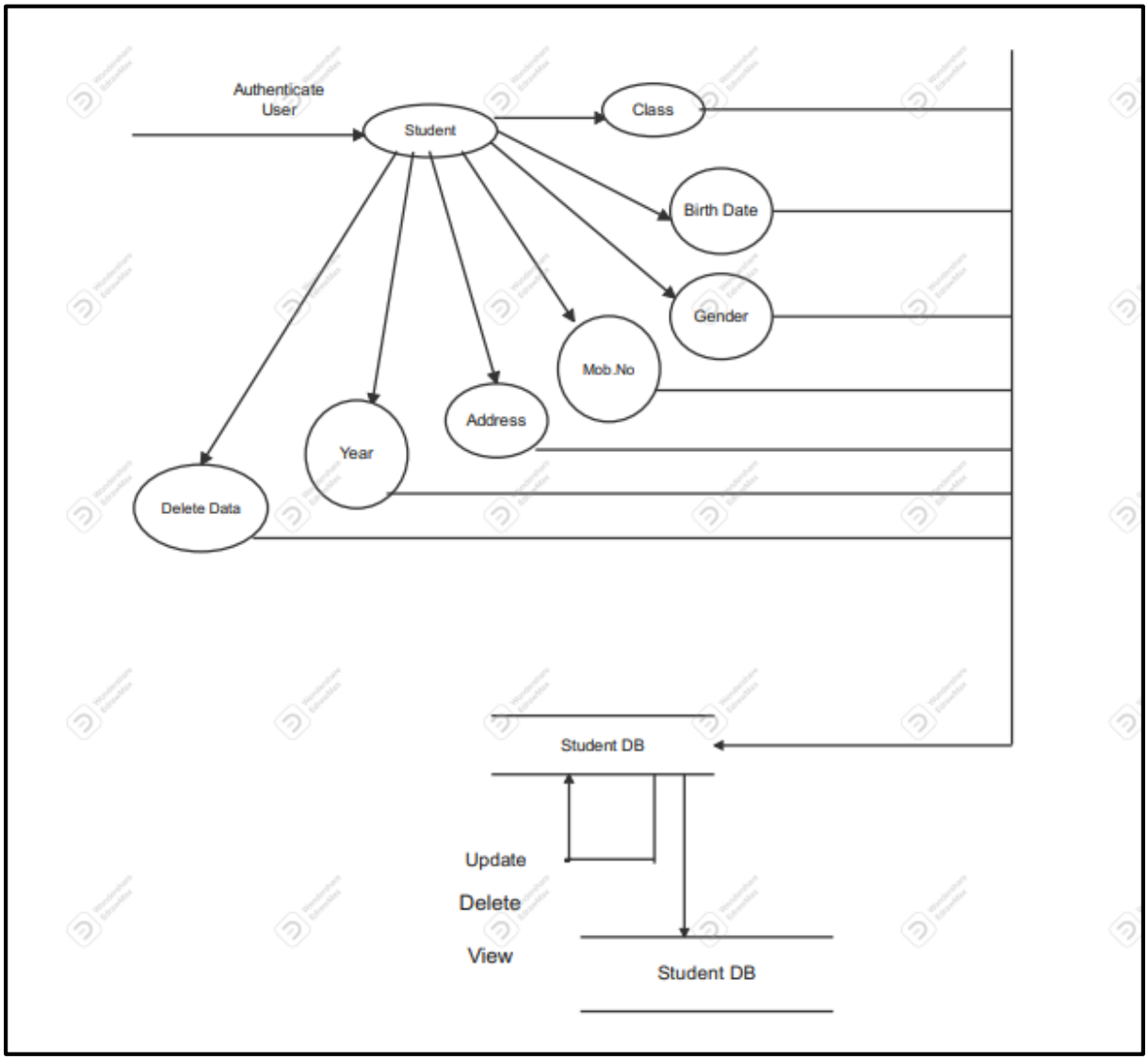
FIRST LEVEL DFD

3. SECOND LEVEL DFD: 2.A:



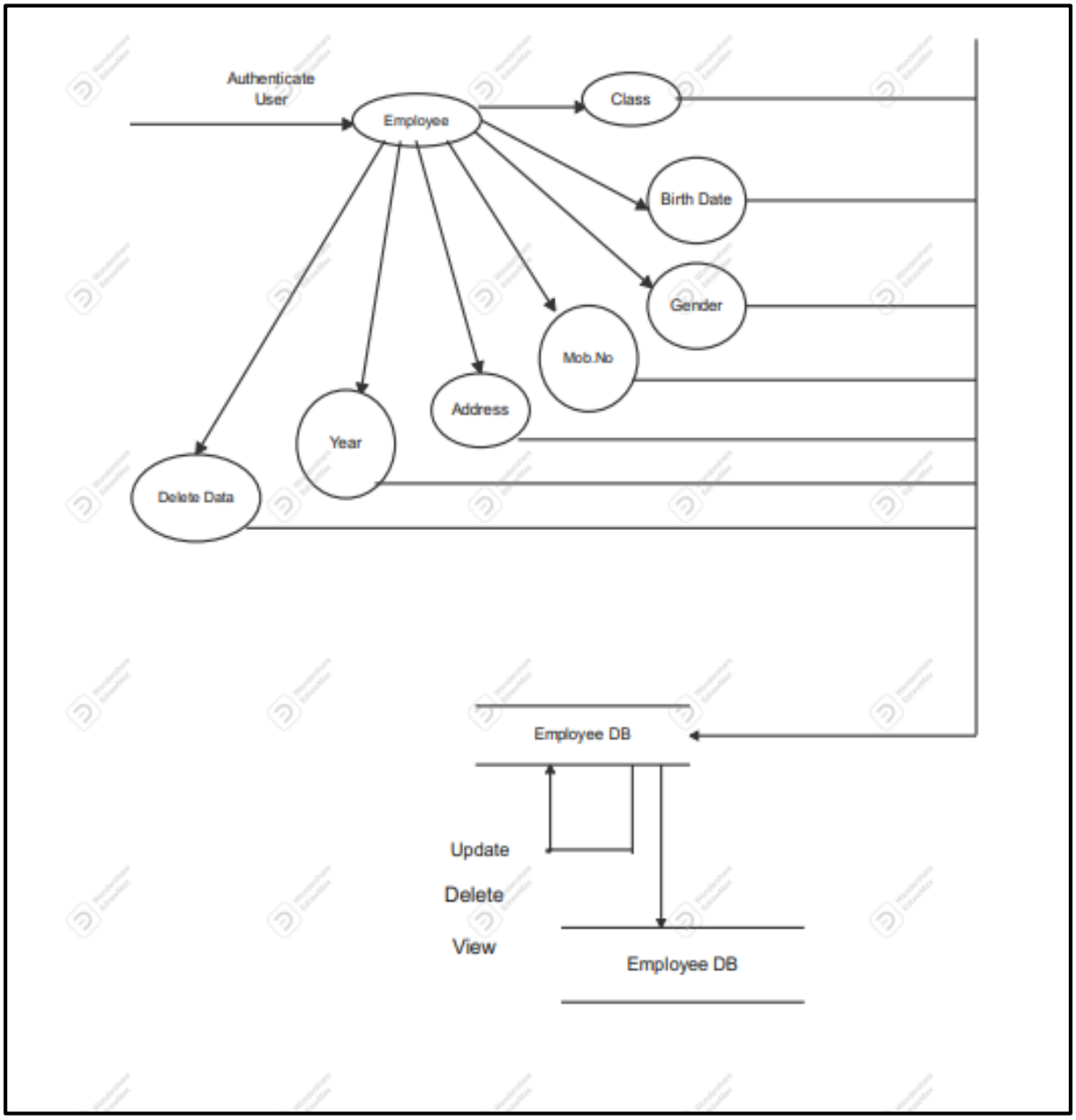
SECOND LEVEL DFD 2.A

3. SECOND LEVEL DFD: 2.B:



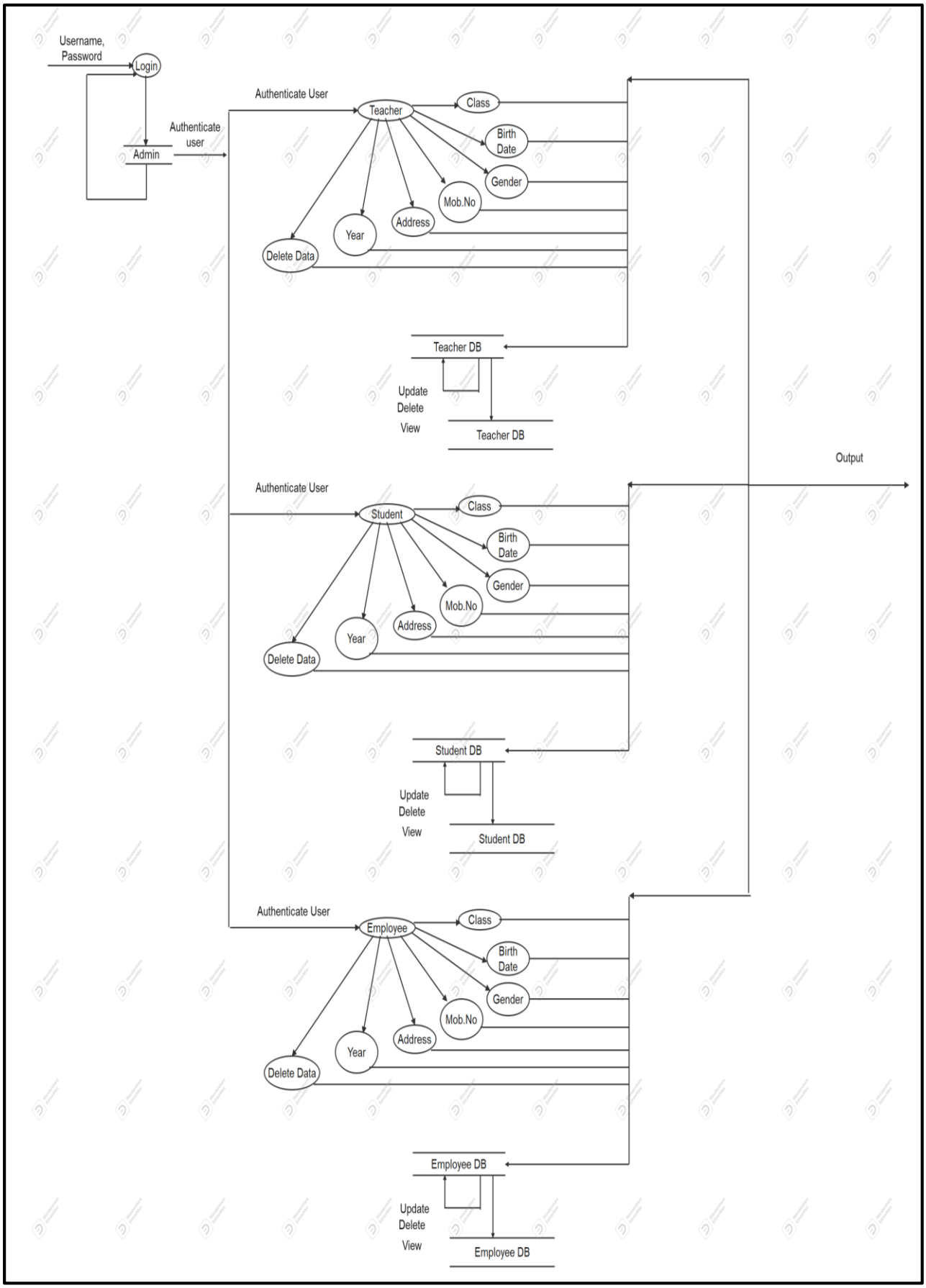
SECOND LEVEL DFD:2.B

3. SECOND LEVEL DFD:2.C:

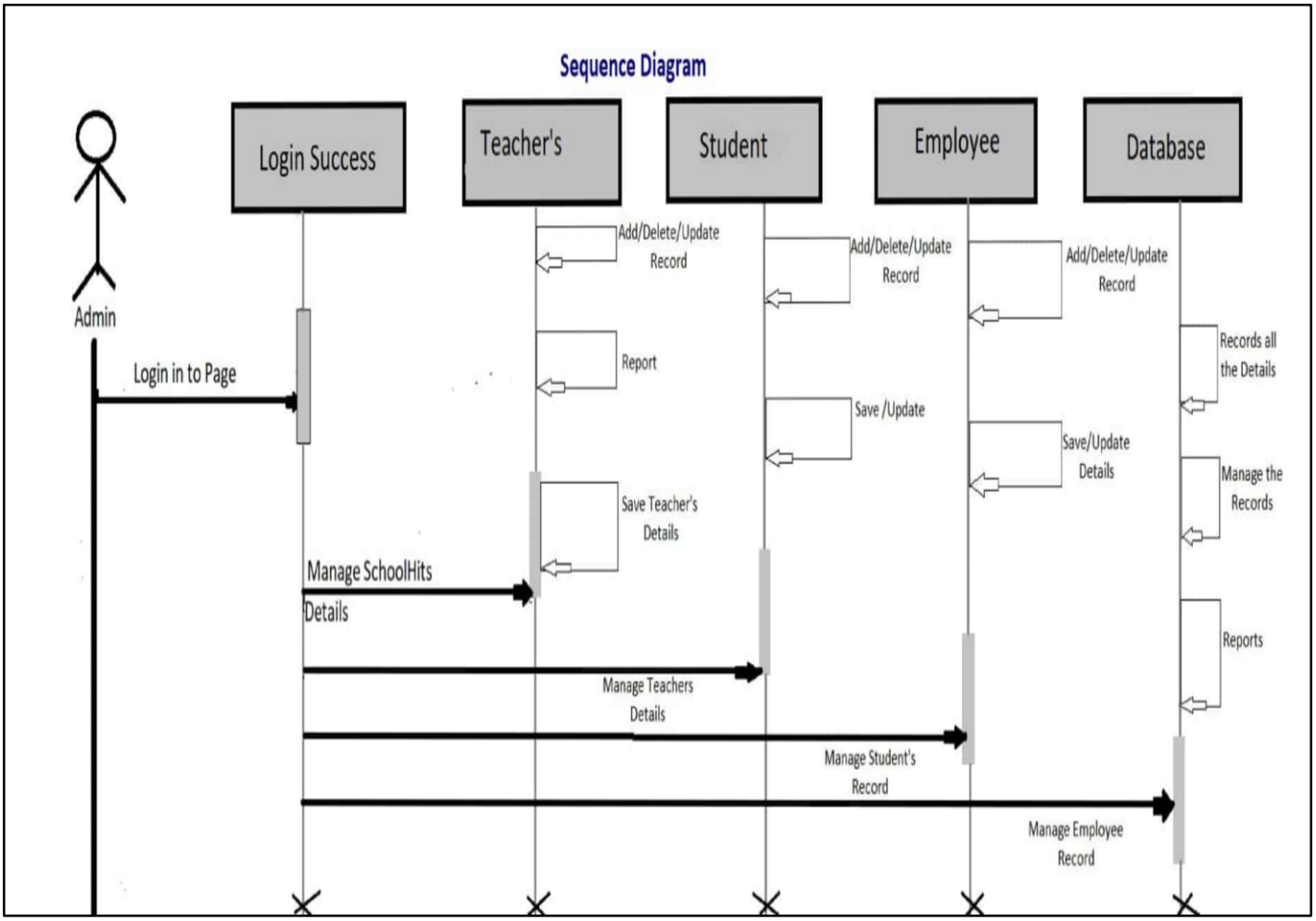


SECOND LEVEL DFD:2.C

4.THIRD LEVEL DFD :



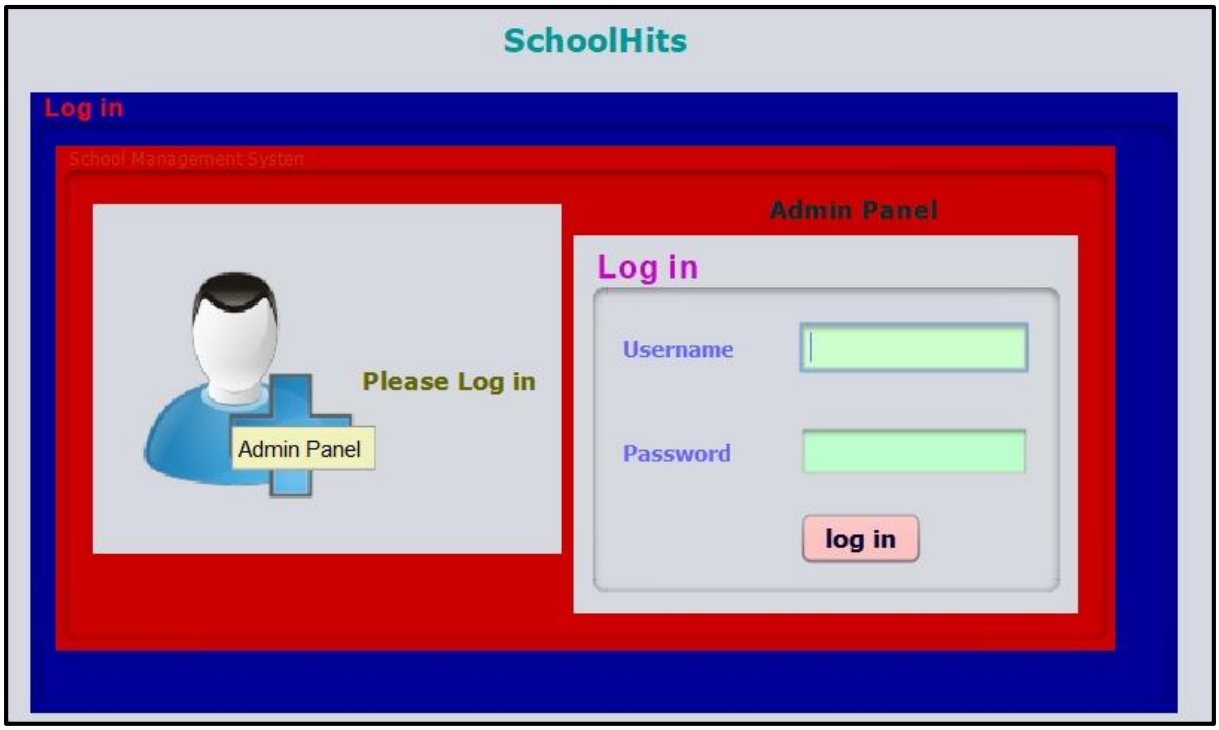
4.3 Sequence Diagram :



Sequence Diagram

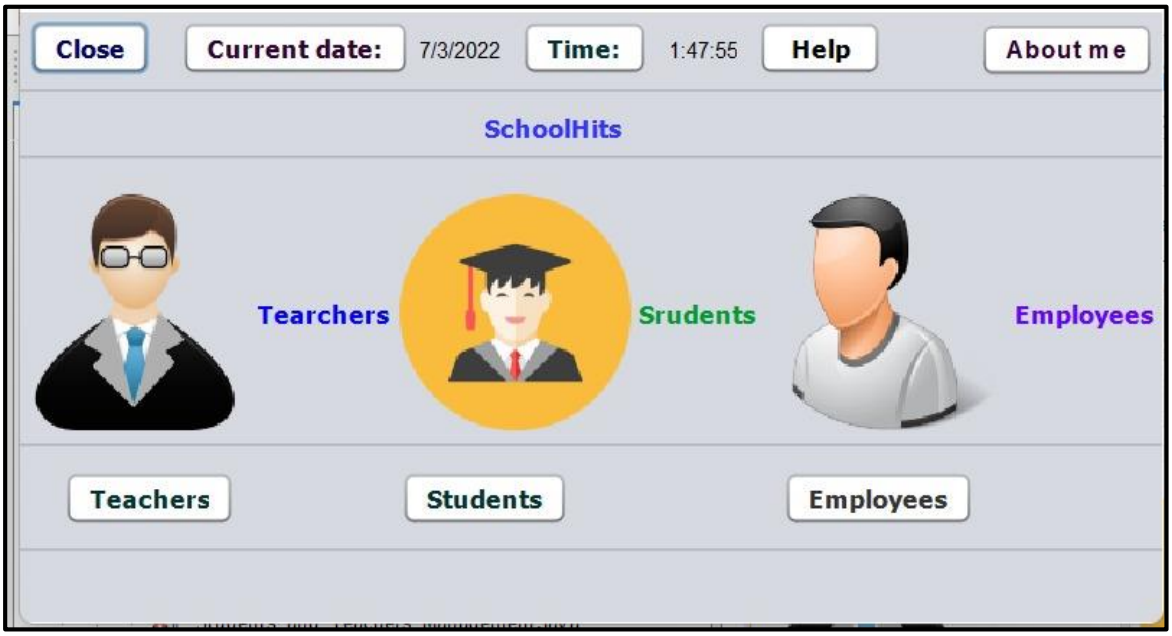
4.4 User Interface Design (Screens etc.):

- **ADMIN LOGIN**



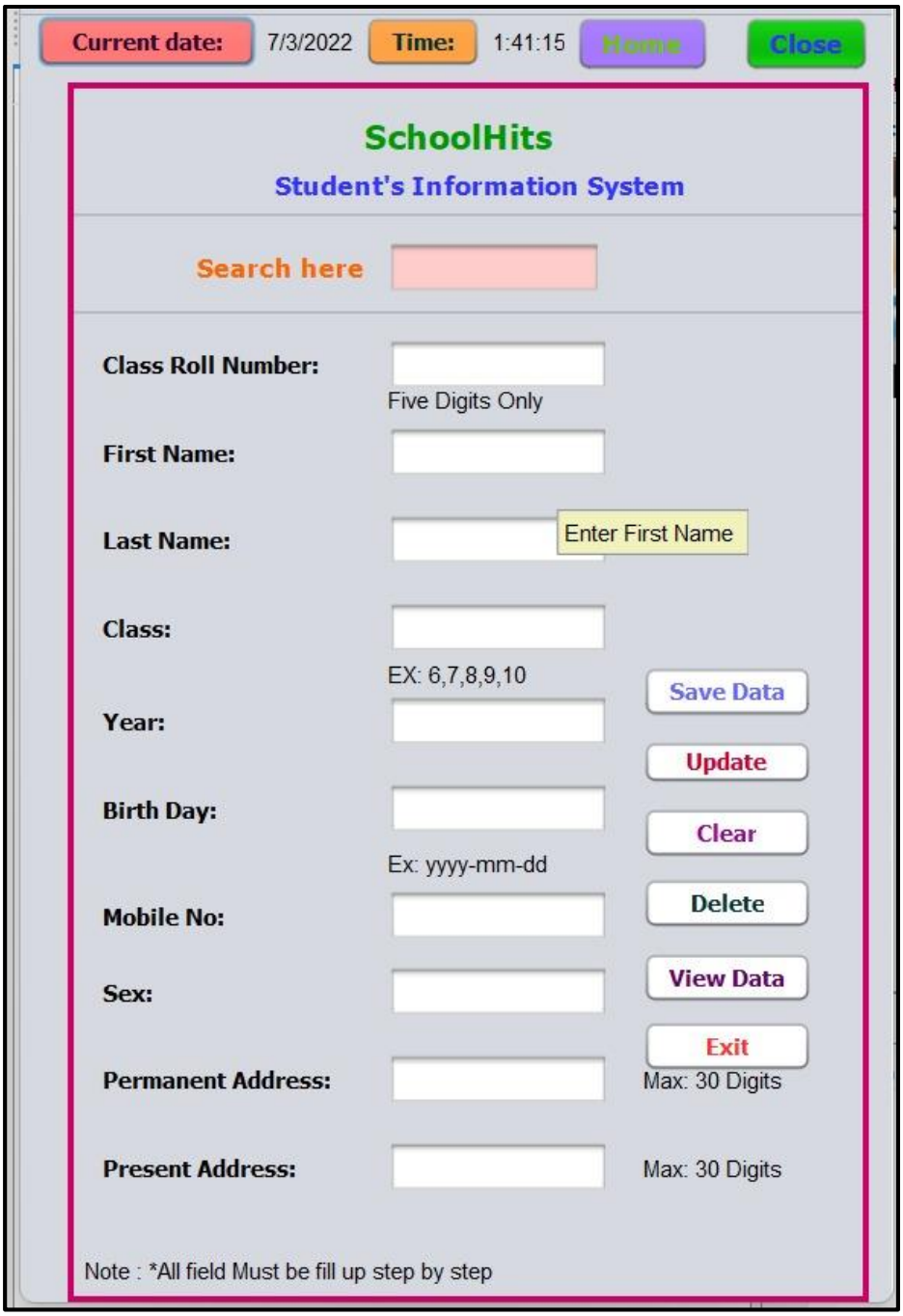
ADMIN LOGIN

➢ INFORMATION SELECTION SECTION:



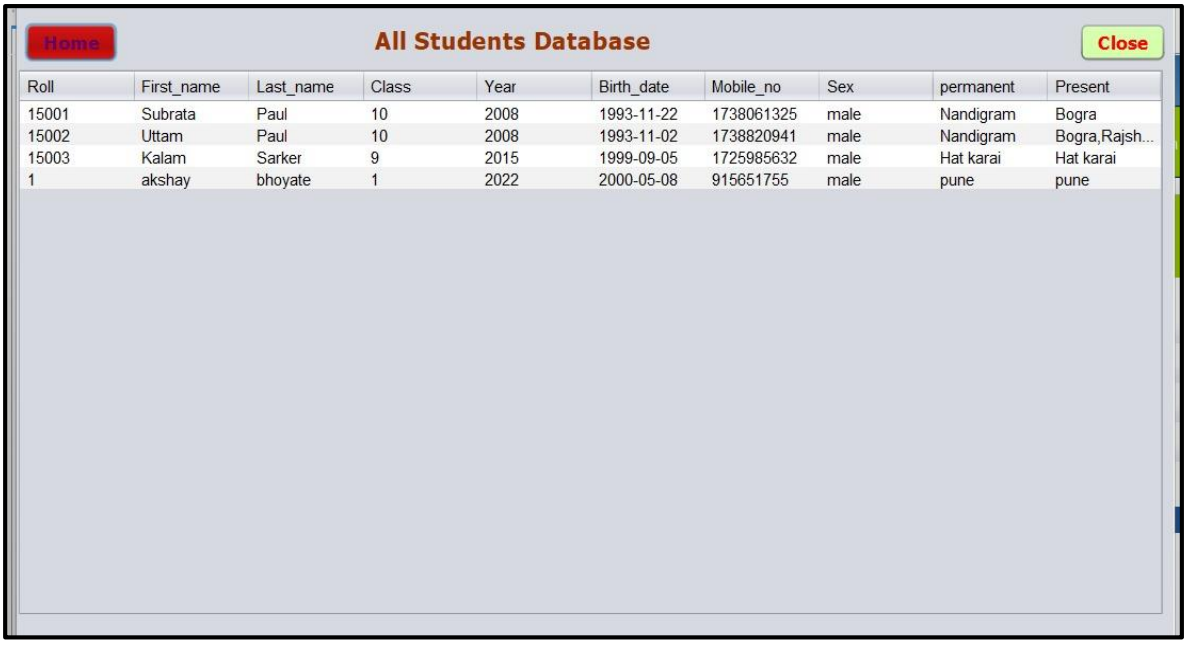
INFORMATION SELECTION SECTION

➢ STUDENT LOGIN:



STUDENT LOGIN

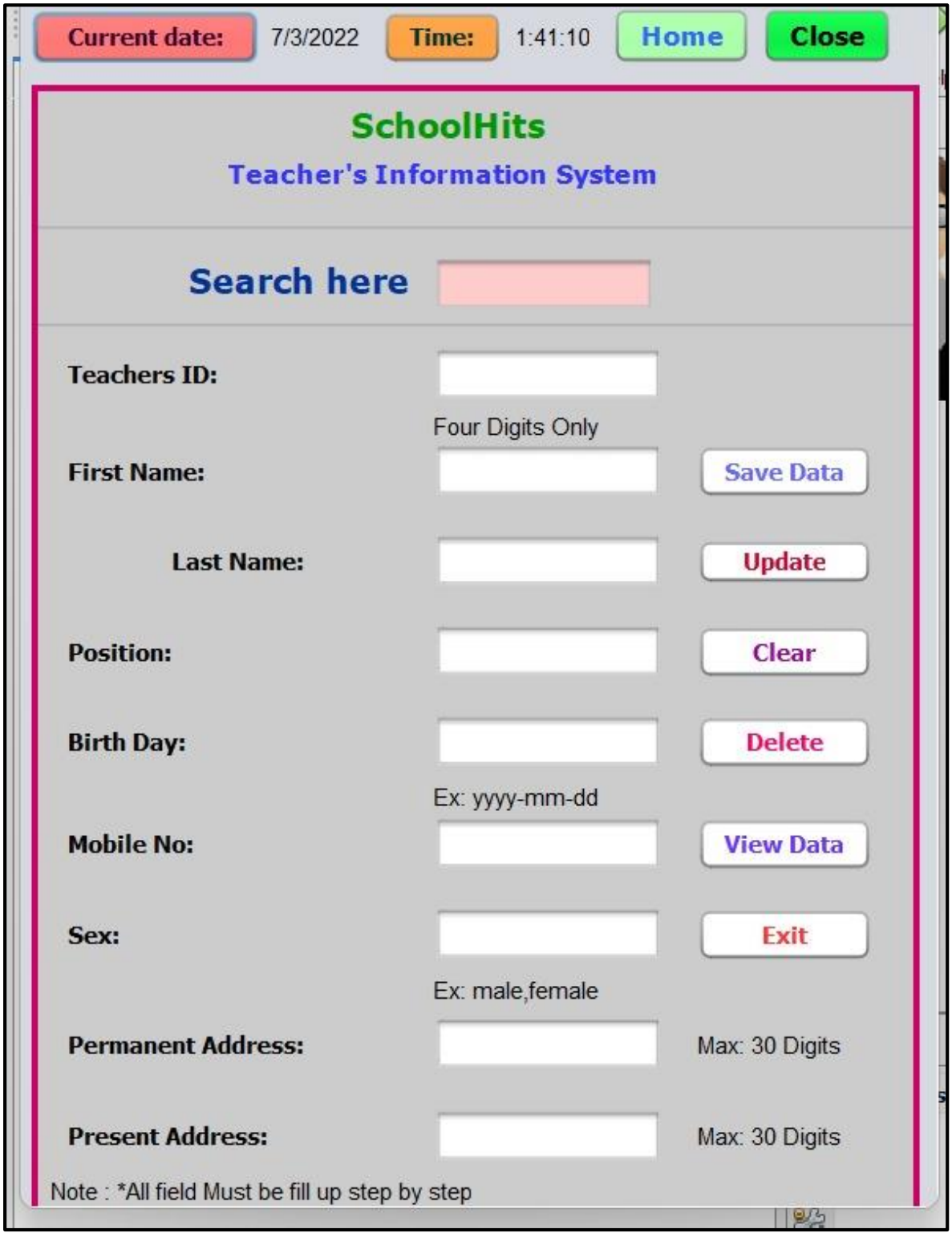
➢ STUDENT DATABASE:



Roll	First_name	Last_name	Class	Year	Birth_date	Mobile_no	Sex	permanent	Present
15001	Subrata	Paul	10	2008	1993-11-22	1738061325	male	Nandigram	Bogra
15002	Uttam	Paul	10	2008	1993-11-02	1738820941	male	Nandigram	Bogra,Rajsh...
15003	Kalam	Sarker	9	2015	1999-09-05	1725985632	male	Hat karai	Hat karai
1	akshay	bhoyate	1	2022	2000-05-08	915651755	male	pune	pune

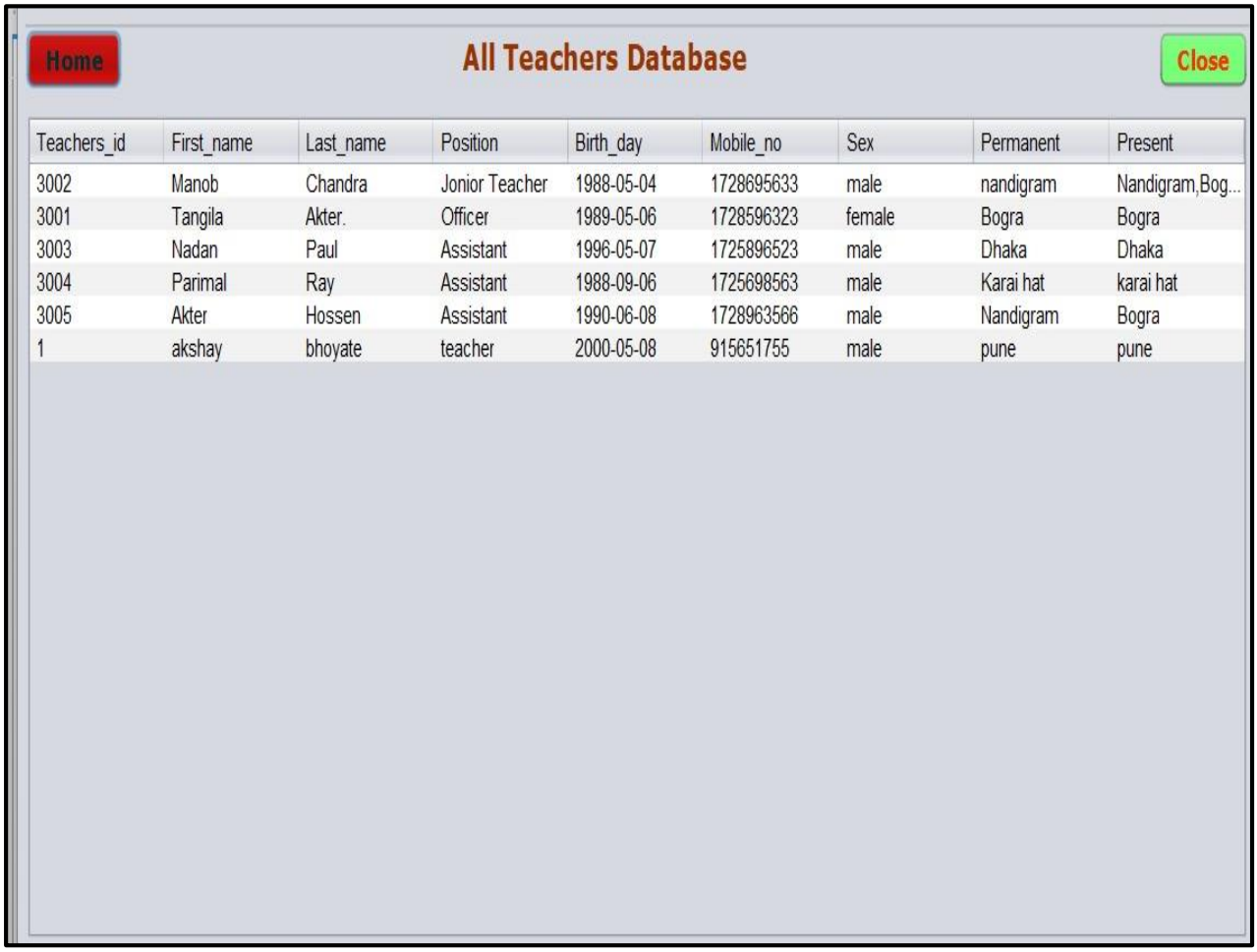
STUDENT DATABASE

➢ TEACHER LOGIN :



TEACHER LOGIN

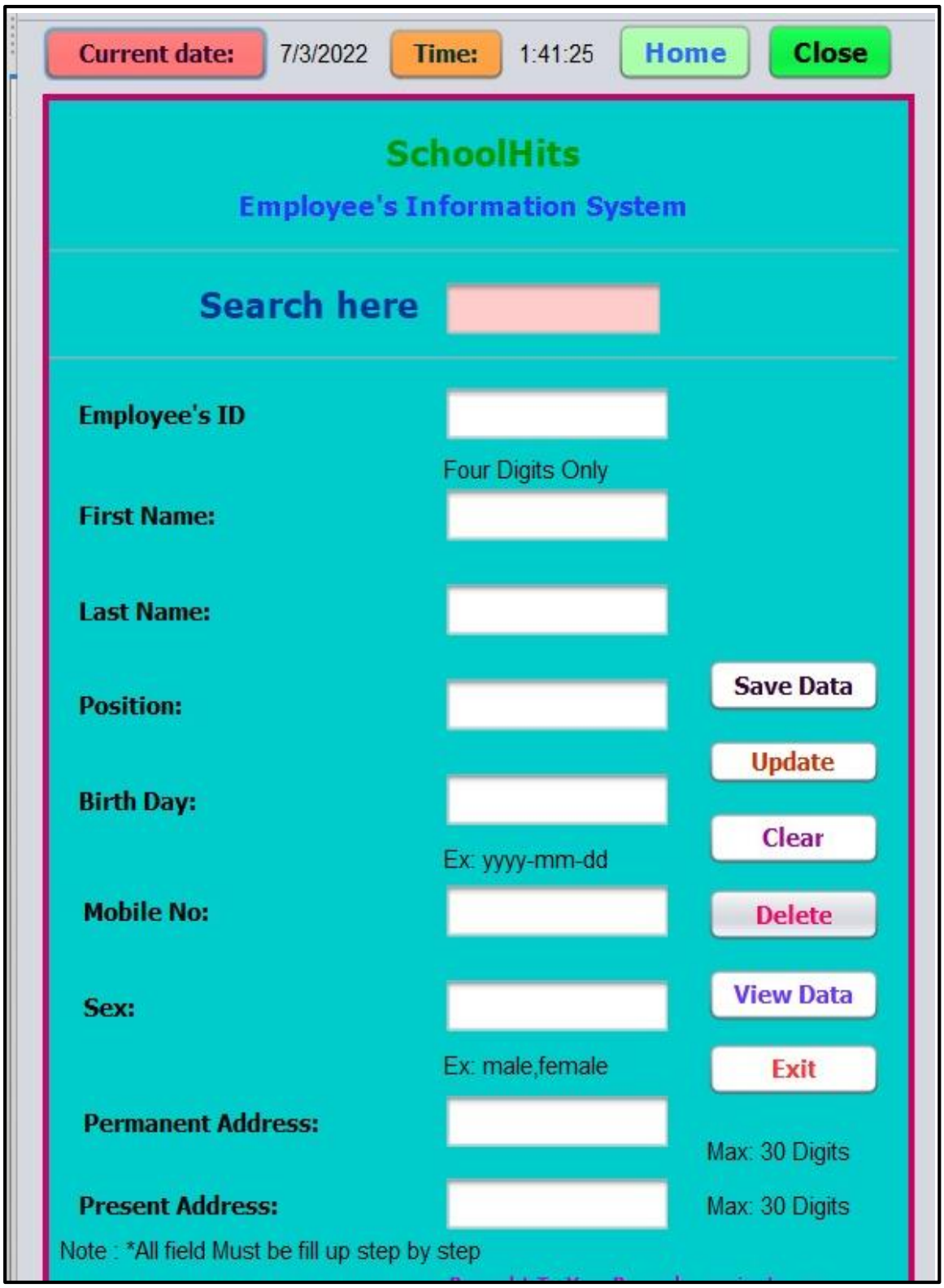
➢ TEACHER DATABASE:



Teachers_id	First_name	Last_name	Position	Birth_day	Mobile_no	Sex	Permanent	Present
3002	Manob	Chandra	Jonior Teacher	1988-05-04	1728695633	male	nandigram	Nandigram,Bog...
3001	Tangila	Akter.	Officer	1989-05-06	1728596323	female	Bogra	Bogra
3003	Nadan	Paul	Assistant	1996-05-07	1725896523	male	Dhaka	Dhaka
3004	Parimal	Ray	Assistant	1988-09-06	1725698563	male	Karai hat	karai hat
3005	Akter	Hossen	Assistant	1990-06-08	1728963566	male	Nandigram	Bogra
1	akshay	bhoyate	teacher	2000-05-08	915651755	male	pune	pune

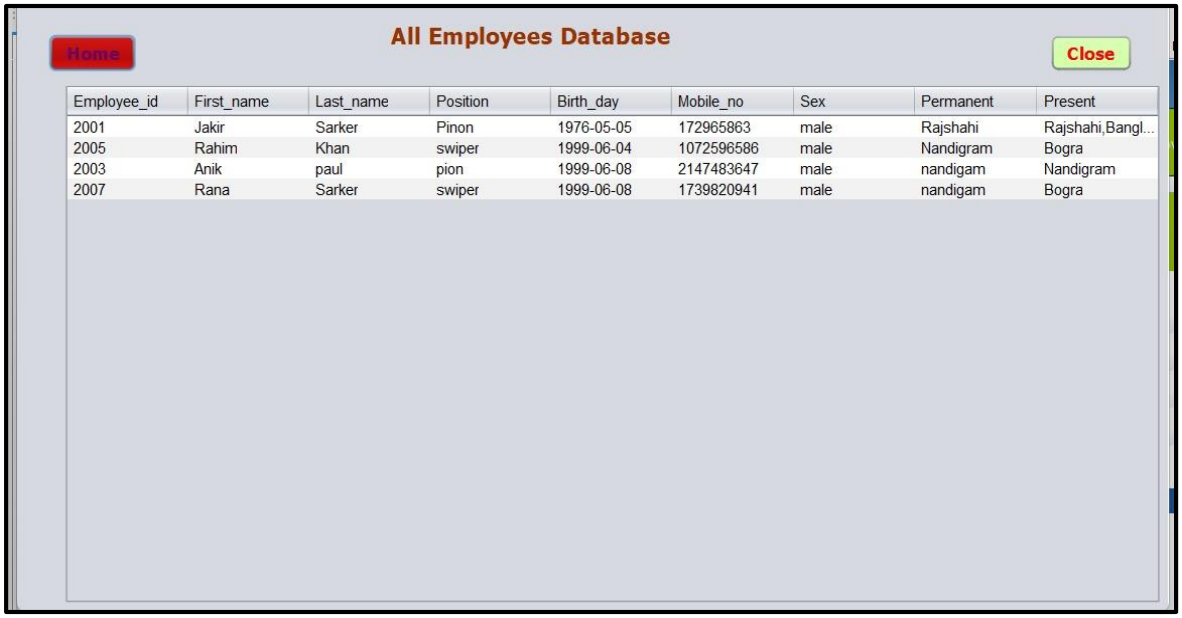
TEACHER DATABASE

➢ EMPLOYEE LOGIN:



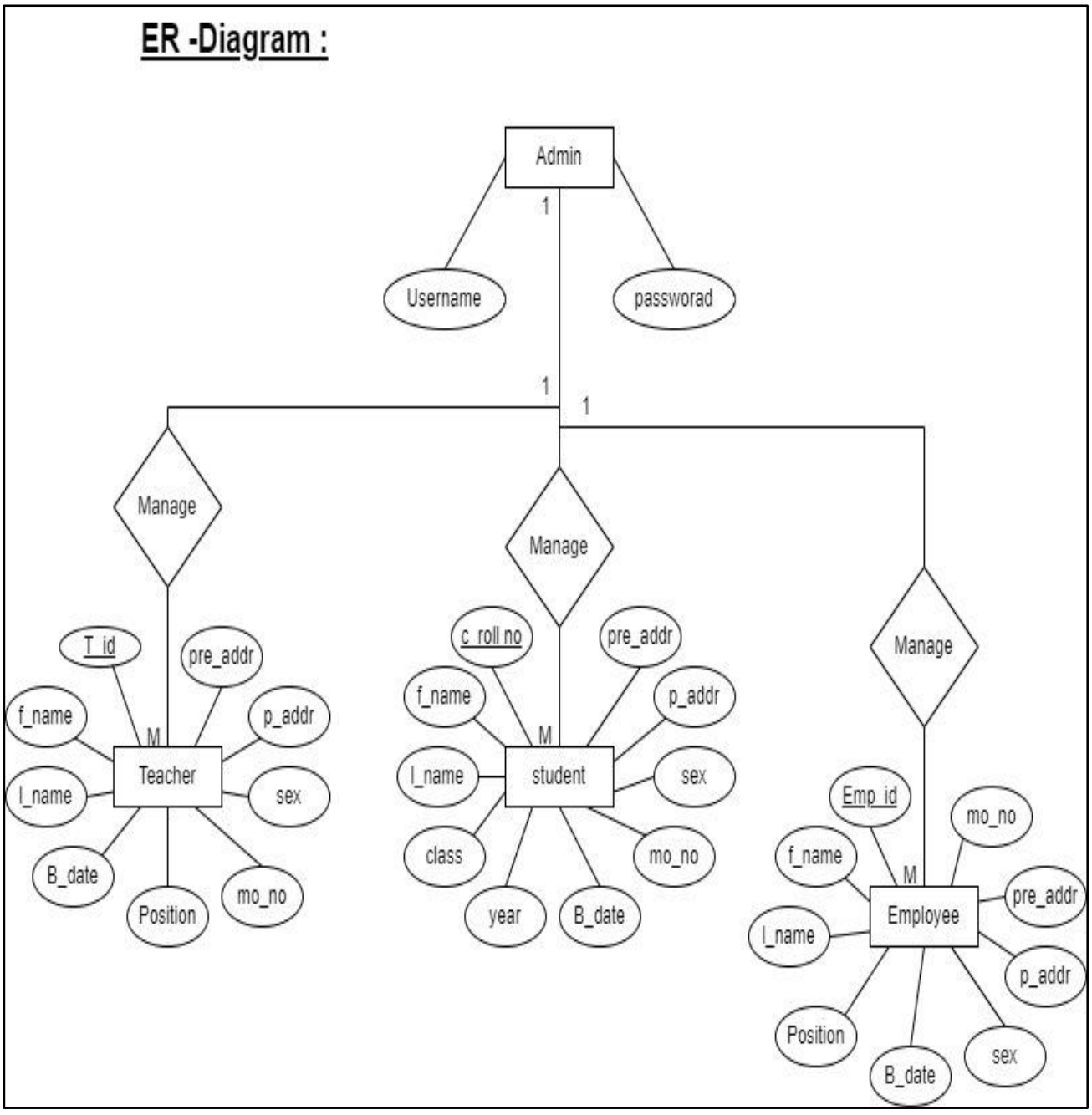
EMPLOYEE LOGIN

➢ EMPLOYEE DATABASE:



EMPLOYEE DATABASE

4.6 ERD (ENTITY RELATIONSHIP DIAGRAM):



ERD (ENTITY RELATIONSHIP DIAGRAM)

4.7 Data Dictionary:

Employee_Data

Employee_id	**Employee_id**	**OT NULL**
`First_name`	**varchar(20**	**NOT NULL,**
`Last_name`	**varchar(20)**	**NOT NULL,**
`Position`	**varchar(20)**	**NOT NULL,**
`Birth_day`	**date**	**NOT NULL,**
`Mobile_no`	**int(20)**	**NOT NULL,**
`Sex`	**varchar(20)**	**NOT NULL,**
`Permanent`	**varchar(50)**	**NOT NULL,**

`Present`	**varchar(50)**	**NOT NULL**
PRIMARY KEY	**(`Employee_id`)**	**--**

Student Data

Employee_id	**Employee_id**	**OT NULL**
`First_name`	**varchar(20**	**NOT NULL,**
`Last_name`	**varchar(20)**	**NOT NULL,**
`Position`	**varchar(20)**	**NOT NULL,**
`Birth_day`	**date**	**NOT NULL,**
`Mobile_no`	**int(20)**	**NOT NULL,**
`Sex`	**varchar(20)**	**NOT NULL,**
`Permanent`	**varchar(50)**	**NOT NULL,**

`Present`	**varchar(50)**	**NOT NULL**
PRIMARY KEY	**(`Employee_id`)**	**--**

Teacher Data

Employee_id	Employee_id	OT NULL
`First_name`	varchar(20	NOT NULL,
`Last_name`	varchar(20)	NOT NULL,
`Position`	varchar(20)	NOT NULL,
`Birth_day`	date	NOT NULL,
`Mobile_no`	int(20)	NOT NULL,
`Sex`	varchar(20)	NOT NULL,
`Permanent`	varchar(50)	NOT NULL,

`Present`	varchar(50)	NOT NULL
PRIMARY KEY	(`Employee_id`)	--

5. Validation Checks:

- The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

- Add user in email should contain @ & .com and contact number should be 10 digits.

- if less than = contact length is less than 10 .
- if greater than 10 = contact length greater than 10.

6. Future scope of the Mini Project:

- This is the most recent school administration framework. Accessible for all kind of schools

- It is accessible with source code.

- The high lights are record beneath Add up to a number of under studies, instructors, administrator, bookkeepers, lodging director, graduated class ,guardians and participation of understudies for that day initially ,Dashboard wise holds a date-book for indicating occasions ,diagrams for different rates of educators ,guardians, under studies participation ,grades ,understudies exhibitions , and so on.

7. CONCLUSION:

- The School Management System which capable of storing the school data.

- It can store the all the data of teacher, student and employees, such as name of student, roll no, class, age, date of birth, mobile no, address, etc.

- We can search the data of any student by their id no. And update the data and save the updated data at the same id no of that student.

- In this system we can track the information of any teacher, student, employees and update or change the information.

8. Bibliography:

- https://www.tutorialspoint.com/index.htm
- https://www.javatpoint.com
- https://www.w3schools.com
- https://html.com

9. ANNEXURES:

9.1:- Sample program code

Admin Page:-

```
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Admin extends javax.swing.JFrame {


    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    /**
     * Creates new form Admin
     */
    public Admin() {
        initComponents();
```

```
        conn=javaconnect.connerDb();
    }


    public  void close()
    {
        WindowEvent winClosingEvent = new
WindowEvent(this, WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEve
nt(winClosingEvent);


    }
    /**
     * This method is called from within the constructor to
initialize the form.
     * WARNING: Do NOT modify this code. The content of
this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
    private void initComponents() {


        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
```

```
        txt_pass = new javax.swing.JPasswordField();
        cmd_admin = new javax.swing.JButton();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


        jLabel1.setText("If are a Admin,you can Delete Only.");


        jLabel2.setText("Please Enter Admin Password");


        jLabel3.setText("Password:");


        cmd_admin.setText("Go");
        cmd_admin.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cmd_adminActionPerformed(evt);
            }
        });


        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(105, 105, 105)
                        .addComponent(jLabel1))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(58, 58, 58)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                            .addComponent(jLabel2)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jLabel3)
                                .addGap(18, 18, 18)
                                .addComponent(txt_pass, javax.swing.GroupLayout.PREFERRED_SIZE, 137, javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addComponent(cmd_admin))))
                .addContainerGap(92, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addComponent(jLabel2)
                .addGap(34, 34, 34)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(txt_pass, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addComponent(cmd_admin)
                .addContainerGap(51, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents
```

```
    private void
cmd_adminActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_cmd_adminActionPerformed
        // TODO add your handling code here:



        String sql="select * from Admin where Password=?";


        try {
            pst=conn.prepareStatement(sql);


            pst.setString(2, txt_pass.getText());


            rs=pst.executeQuery();
            if(rs.next()){
                JOptionPane.showMessageDialog(null, "Log in
Successful.");
                rs.close();
                pst.close();


                close();


            }
            else{
                JOptionPane.showMessageDialog(null, "You Enter
Worng Password.");
```

```
            }
        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }



    }//GEN-LAST:event_cmd_adminActionPerformed


    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
```

```
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName
());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Admin.class.getName()).l
og(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Admin.class.getName()).l
og(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Admin.class.getName()).l
og(java.util.logging.Level.SEVERE, null, ex);
        } catch
(javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Admin.class.getName()).l
og(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
```

```
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Admin().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-
BEGIN:variables
    private javax.swing.JButton cmd_admin;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPasswordField txt_pass;
    // End of variables declaration//GEN-END:variables
}
```

THANK YOU!!!!